

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	379	(715/511).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/06/18 09:19
L2	1143	(715/530).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/06/18 09:19
L3	182	(715/509).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/06/18 09:19
L4	1471	(707/203).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/06/18 09:19
L5	592	sql with column with query	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L6	8204	obtain\$4 adj2 column	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L7	21	L5 and L6	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L8	293	sql with advantage	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L9	2	L7 and L8	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L10	506	associat\$4 with version with number with data	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L11	1626	stor\$4 with version with table	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L12	270	stor\$4 with version with table with (current or new or recent)	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L13	42	stor\$4 with version with table with old	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L14	23	12 and 13	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58
L15	23	14 and 11	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L16	2	14 and 10	US-PGPUB; USPAT	OR	ON	2006/06/18 10:56
L17	184	database with old with version	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58
L18	1048	database with new with version	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58
L19	118	17 and 18	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58
L20	27973	version with number	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58
L21	72	19 and 20	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58

EAST Search History

L22	66	21 and value	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58
L23	48	21 and (version with value)	US-PGPUB; USPAT	OR	ON	2006/06/18 10:58

	Document ID	Issue Date	Title	Current OR
1	US 20060109807 A1	20060525	Multicasting using tunneling method	370/315
2	US 20050216520 A1	20050929	Systems and methods for versioning based triggers	707/200
3	US 20050210460 A1	20050922	Computing device with relatively limited storage space and operating/file system thereof	717/168
4	US 20050210076 A1	20050922	Computing device with relatively limited storage space and operating/file system thereof	707/200
5	US 20050210055 A1	20050922	Computing device with relatively limited storage space and operating/file system thereof	707/101
6	US 20050209991 A1	20050922	Computing device with relatively limited storage space and operating / file system thereof	707/1
7	US 20050203969 A1	20050915	Version management system and version management method for content delivery and management	707/203
8	US 20040117358 A1	20040617	Method, system, and program for an improved enterprise spatial system	707/3

	Document ID	Issue Date	Title	Current OR
9	US 20040111341 A1	20040610	Electronic data transaction method and electronic data transaction system	705/35
10	US 20030061245 A1	20030327	Implementing versioning support for data using a two-table approach that maximizes database efficiency	707/203
11	US 20030037323 A1	20030220	Method for upgrading data	717/170
12	US 20020138497 A1	20020926	Method, system, and program for implementing a database trigger	707/104.1
13	US 20010054042 A1	20011220	COMPUTING SYSTEM FOR INFORMATION MANAGEMENT	707/100
14	US 20010002914 A1	20010607	Method of distributing program to a plurality of nodes within a network by using gateway	370/535
15	US 6980558 B2	20051227	Method of distributing program to a plurality of nodes within a network by using gateway	370/401
16	US 6952717 B1	20051004	Document and message exchange system for ASP model	709/205
17	US 6829616 B2	20041207	Method, system, and program for implementing a database trigger	707/102

	Document ID	Issue Date	Title	Current OR
18	US 6738801 B1	20040518	Master server facilitating communication between slave servers in incompatible data formats, and method for upgrading slave servers	709/208
19	US 6457017 B2	20020924	Computing system for information management	707/103R
20	US 6071317 A	20000606	Object code logic analysis and automated modification system and method	717/128
21	US 5859977 A	19990112	System for software update in manner based on processing properties of devices via maintenance network while allowing data transmission between devices on another network	709/223
22	US 5758156 A	19980526	Method and apparatus of testing programs	713/100
23	US 5495612 A	19960227	System for dynamically changing an execution program and method for doing the same	719/331


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used version number value table new old database

Found 123,704 of 178,880

Sort results by


[Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

1 [A structured approach for the definition of the semantics of active databases](#)



Piero Fraternali, Letizia Tanca

 December 1995 **ACM Transactions on Database Systems (TODS)**, Volume 20 Issue 4

Publisher: ACM Press

Full text available: pdf(4.15 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Active DBMSs couple database technology with rule-based programming to achieve the capability of reaction to database (and possibly external) stimuli, called events. The reactive capabilities of active databases are useful for a wide spectrum of applications, including security, view materialization, integrity checking and enforcement, or heterogeneous database integration, which makes this technology very promising for the near future. An active database system consists of ...

Keywords: active database systems, database rule processing, events, fixpoint semantics, rules, semantics

2 [TransformGen: automating the maintenance of structure-oriented environments](#)



David Garlan, Charles W. Krueger, Barbara Staudt Lerner

 May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 16 Issue 3

Publisher: ACM Press

Full text available: pdf(3.10 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

A serious problem for programs that use persistent data is that information created and maintained by the program becomes invalid if the persistent types used in the program are modified in a new release. Unfortunately, there has been little systematic treatment of the problem; current approaches are manual, ad hoc, and time consuming both for programmers and users. In this article we present a new approach. Focusing on the special case of managing abstract syntax trees in structure-oriented ...

Keywords: schema evolution, structure-oriented environments, type evolution

3 [A linear-time scheme for version reconstruction](#)

Lin Yu, Daniel J. Rosenkrantz

 May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,

Volume 16 Issue 3

**Publisher:** ACM Press

Full text available: pdf(1.47 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

An efficient scheme to store and reconstruct versions of sequential files is presented. The reconstruction scheme involves building a data structure representing a complete version, and then successively modifying this data structure by applying a sequence of specially formatted differential files to it. Each application of a differential file produces a representation of an intermediate version, with the final data structure representing the requested version. The scheme uses a ...

Keywords: data structures, database systems, differential files, document preparation, software systems, textual objects, version control

4 Semantic integrity support in SQL:1999 and commercial (object-)relational database management systems

Can Türker, Michael Gertz

December 2001 **The VLDB Journal — The International Journal on Very Large Data****Bases**, Volume 10 Issue 4**Publisher:** Springer-Verlag New York, Inc.Full text available: pdf(345.55 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The correctness of the data managed by database systems is vital to any application that utilizes data for business, research, and decision-making purposes. To guard databases against erroneous data not reflecting real-world data or business rules, semantic integrity constraints can be specified during database design. Current commercial database management systems provide various means to implement mechanisms to enforce semantic integrity constraints at database run-time. In this paper, we give ...

Keywords: Constraint enforcement, Object-relational databases, SQL:1999, Semantic integrity constraints

5 Databases: Testing database transactions with AGENDA



Yuetang Deng, Phyllis Frankl, David Chays

May 2005 **Proceedings of the 27th international conference on Software engineering****Publisher:** ACM PressFull text available: pdf(185.68 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

AGENDA is a tool set for testing relational database applications. An earlier prototype was targeted to applications consisting of a single query and included components for populating a database with data suitable for testing the application, generating inputs to the query, and checking relatively simple aspects of the results of executing the query. This paper describes substantial extensions to AGENDA, allowing it to test transactions with multiple queries and with complex intended behavior. ...

Keywords: database, software testing, transaction

6 Dynamic software updating



Michael Hicks, Scott Nettles

November 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 6**Publisher:** ACM PressFull text available: pdf(622.69 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many important applications must run continuously and without interruption, and yet also must be changed to fix bugs or upgrade functionality. No prior general-purpose methodology for dynamic updating achieves a practical balance between flexibility, robustness, low overhead, ease of use, and low cost. We present an approach for C-like languages that provides type-safe dynamic updating of native code in an extremely flexible manner---code, data, and types may be updated, at programmer-determined ...

Keywords: Dynamic software updating, typed assembly language

7 Paper session DB-5 (databases): updates and change detection: Detecting changes on unordered XML documents using relational databases: a schema-conscious approach

Erwin Leonardi, Sourav S. Bhowmick

October 2005 **Proceedings of the 14th ACM international conference on Information and knowledge management CIKM '05**

Publisher: ACM Press

Full text available:  pdf(307.12 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Several *relational approaches* have been proposed to detect the changes to XML documents by using relational databases. These approaches store the XML documents in the relational database and issue SQL queries (whenever appropriate) to detect the changes. All of these relational-based approaches use the *schema-oblivious* XML storage strategy for detecting the changes. However, there is growing evidence that schema-conscious storage approaches perform significantly better than schema- ...

Keywords: XML, change detection, change management

8 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Publisher: IBM Press

Full text available:  pdf(4.21 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

9 Special issue on spatial database systems: Management of multidimensional discrete data

Peter Baumann

October 1994 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 3 Issue 4

Publisher: Springer-Verlag New York, Inc.

Full text available:  pdf(2.30 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Spatial database management involves two main categories of data: vector and raster data. The former has received a lot of in-depth investigation; the latter still lacks a sound framework. Current DBMSs either regard raster data as pure byte sequences where the DBMS has no knowledge about the underlying semantics, or they do not complement array structures with storage mechanisms suitable for huge arrays, or they are designed as specialized systems with sophisticated imaging functionality, but n ...

Keywords: Multimedia database systems, image database systems, spatial index, tiling

10 An efficient I/O interface for optical disks



Jeffrey S. Vitter

June 1985 **ACM Transactions on Database Systems (TODS)**, Volume 10 Issue 2

Publisher: ACM Press

Full text available: [pdf\(2.55 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We introduce the notion of an I/O interface for optical digital (write-once) disks, which is quite different from earlier research. The purpose of an I/O interface is to allow existing operating systems and application programs that use magnetic disks to use optical disks instead, with minimal change. We define what it means for an I/O interface to be disk-efficient. We demonstrate a practical disk-efficient I/O interface and show that its I/O performance in many cases is optimum, up to a ...

11 Recovery Techniques for Database Systems



Joost S. M. Verhofstad

June 1978 **ACM Computing Surveys (CSUR)**, Volume 10 Issue 2

Publisher: ACM Press

Full text available: [pdf\(2.32 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

12 Version models for software configuration management



Reidar Conradi, Bernhard Westfechtel

June 1998 **ACM Computing Surveys (CSUR)**, Volume 30 Issue 2

Publisher: ACM Press

Full text available: [pdf\(483.54 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

After more than 20 years of research and practice in software configuration management (SCM), constructing consistent configurations of versioned software products still remains a challenge. This article focuses on the version models underlying both commercial systems and research prototypes. It provides an overview and classification of different versioning paradigms and defines and relates fundamental concepts such as revisions, variants, configurations, and changes. In particular, we foc ...

Keywords: changes, configuration rules, configurations, revisions, variants, versions

13 System R: relational approach to database management



M. M. Astrahan, M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. N. Gray, P. P. Griffiths, W. F. King, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, V. Watson

June 1976 **ACM Transactions on Database Systems (TODS)**, Volume 1 Issue 2

Publisher: ACM Press

Full text available: [pdf\(3.18 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

System R is a database management system which provides a high level relational data interface. The system provides a high level of data independence by isolating the end user as much as possible from underlying storage structures. The system permits definition of a variety of relational views on common underlying data. Data control features are provided, including authorization, integrity assertions, triggered transactions,

a logging and recovery subsystem, and facilities for maintaining ...

Keywords: authorization, data structures, database, index structures, locking, nonprocedural language, recovery, relational model

14 Comparison of access methods for time-evolving data



Betty Salzberg, Vassilis J. Tsotras

June 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 2

Publisher: ACM Press

Full text available: pdf(529.53 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper compares different indexing techniques proposed for supporting efficient access to temporal data. The comparison is based on a collection of important performance criteria, including the space consumed, update processing, and query time for representative queries. The comparison is based on worst-case analysis, hence no assumptions on data distribution or query frequencies are made. When a number of methods have the same asymptotic worst-case behavior, features in the methods tha ...

Keywords: I/O performance, access methods, structures, temporal databases

15 ProbView: a flexible probabilistic database system



Laks V. S. Lakshmanan, Nicola Leone, Robert Ross, V. S. Subrahmanian

September 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 3

Publisher: ACM Press

Full text available: pdf(1.92 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Probability theory is mathematically the best understood paradigm for modeling and manipulating uncertain information. Probabilities of complex events can be computed from those of basic events on which they depend, using any of a number of strategies. Which strategy is appropriate depends very much on the known interdependencies among the events involved. Previous work on probabilistic databases has assumed a fixed and restrictive combination strategy (e ...

Keywords: probabilistic databases, view maintenance

16 Query processing techniques in the summary-table-by-example database query language



Gultekin Özsoyoğlu, Victor Matos, Meral Özsoyoğlu

December 1989 **ACM Transactions on Database Systems (TODS)**, Volume 14 Issue 4

Publisher: ACM Press

Full text available: pdf(3.52 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Summary-Table-by-Example (STBE) is a graphical language suitable for statistical database applications. STBE queries have a hierarchical subquery structure and manipulate summary tables and relations with set-valued attributes. The hierarchical arrangement of STBE queries naturally implies a tuple-by-tuple subquery evaluation strategy (similar to the nested loops join implementation technique) which may not be the best query processing strategy. In this paper we discuss the query ...

17

Principles of transaction-oriented database recovery



Theo Haerder, Andreas Reuter

December 1983 **ACM Computing Surveys (CSUR)**, Volume 15 Issue 4**Publisher:** ACM Press

Full text available: pdf(2.48 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#), [review](#)**18** [Beyond schema evolution to database reorganization](#)

Barbara Staudt Lerner, A. Nico Habermann

September 1990 **ACM SIGPLAN Notices , Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications OOPSLA/ECOOP '90**, Volume 25 Issue 10**Publisher:** ACM Press

Full text available: pdf(979.63 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

While the contents of databases can be easily changed, their organization is typically extremely rigid. Some databases relax the rigidity of database organization somewhat by supporting simple changes to individual schemas. As described in this paper, OTGen supports not only more complex schema changes, but also database reorganization. A database administrator uses a declarative notation to describe mappings between objects created with old versions of schemas and their corresponding repre ...

19 [An algebraic approach to static analysis of active database rules](#)

Elena Baralis, Jennifer Widom

September 2000 **ACM Transactions on Database Systems (TODS)**, Volume 25 Issue 3**Publisher:** ACM Press

Full text available: pdf(391.93 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Rules in active database systems can be very difficult to program due to the unstructured and unpredictable nature of rule processing. We provide static analysis techniques for predicting whether a given rule set is guaranteed to terminate and whether rule execution is confluent (guaranteed to have a unique final state). Our methods are based on previous techniques for analyzing rules in active database systems. We improve considerably on the previous techniques by providing analysis criterion ...

Keywords: active database systems, confluence, database rule processing, database trigger processing, termination

20 [Concurrency Control in Distributed Database Systems](#)

Philip A. Bernstein, Nathan Goodman

June 1981 **ACM Computing Surveys (CSUR)**, Volume 13 Issue 2**Publisher:** ACM Press

Full text available: pdf(3.24 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)